

Fuzzy Motion Adaptive Algorithm for Video De-interlacing

P. Brox¹, I. Baturone¹, S. Sánchez-Solano¹, J. Gutiérrez-Ríos² and F. Fernández-Hernández²

¹ Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC)
Avda. Reina Mercedes S/N. Edificio CICA. 41012 Sevilla-SPAIN
e-mail:brox@imse.cnm.es

² Dpto. Tecnología Fotónica. Universidad Politécnica de Madrid
Campus de Montegancedo S/N. 28660 Boadilla del Monte-Madrid-SPAIN
e-mail:jgr@fi.upm.es

Abstract. A motion adaptive algorithm for video de-interlacing is presented in this paper. It is based on a fuzzy inference system, which performs an interpolation between two linear techniques as a function of the motion level. Fuzzy systems with different number of 'if-then' rules have been analyzed and compared in terms of complexity as well as efficiency in de-interlacing benchmark video sequences.

Key words: Video De-interlacing, Motion Adaptive, Fuzzy Inference Systems, Supervised Learning Algorithms

1 Introduction

The main video transmission formats of TV signals (NTSC, PAL, SECAM) use an interlaced signal, where only half of the lines which compose a frame are transmitted. Therefore, the bandwidth required by the broadcast is effectively halved since the human visual system is less sensitive to flickering of details than to large areas flicker [1]. However, the need of progressive scanning is growing nowadays due to the advent of high-definition television, videophone, projectors, DVDs, and video on PCs. This increasing need has encouraged the development of algorithms that perform a spatio-temporal sampling to calculate the non-transmitted lines.

Among the de-interlacing algorithms two categories can be distinguished: motion-compensated algorithms that use a motion vector to interpolate the missing lines, and non-motion-compensated algorithms [2]. The first ones generally perform better than the second ones especially for sequences with a high level of motion. Unfortunately, the motion-compensated algorithms involve the high computational cost related to motion vector calculation. The different de-interlacing algorithms can be classified by considering if they always interpolate the same pixels (linear techniques) [3], [4], or if these pixels are selected accordingly to the characteristics of the image (non-linear techniques) [5], [6].

Non-linear techniques can be divided in turn into two groups: those which try to adapt the interpolation strategy to the presence of motion [5]; and other ones that perform an edge-dependent interpolation [6].

To implement correctly the motion adaptive algorithm, it is fundamental to detect motion accurately. Basically motion detectors evaluate the difference between luminance values of pixels from two consecutive fields. However, this measurement is not usually very reliable due to the presence of edges, vertical details, and noise corrupting the TV signal. The robustness of these detectors can be increased by using more than one detector, and combining them with the logical operator 'and' [1]. In this way, only if all of them detect motion the motion signal is activated. Other authors resort to the use of a multilevel signal, rather than a binary one, to indicate the probability of motion. Several algorithms based on fuzzy-logic have also been proposed to perform an adaptive interpolation with the level of motion. They exploit the capacity of fuzzy techniques to perform interpolations where the information is uncertain and, hence, the decision is not trivial [7]. The technique proposed in [7] provides good results but it uses a complex set of rules, which requires a considerable computational cost.

A novel motion adaptive algorithm for video de-interlacing is proposed in this paper. It uses a fuzzy logic-based system to determine the interpolation between the pixels from the transmitted lines accordingly to the level of motion. The algorithm is described in detail in Section 2. Its performance when de-interlacing several image sequences is analyzed in Section 3. Finally, concluding remarks are included in Section 4.

2 Algorithm Description

The fuzzy motion adaptive algorithm is based on the following heuristic knowledge: if the pixel to interpolate belongs to an area where there is no motion, the best result is achieved by performing an interpolation among pixels from the previous field (temporal interpolation); however, in the case that the pixel corresponds to an area with a high level of motion then the best solution is to realize an interpolation among pixels from the current field (spatial interpolation). The most basic interpolations have been selected among spatial and temporal linear techniques: pixel insertion from the previous field as temporal interpolation (I_T) and the average value of the pixel from the upper and lower lines as spatial technique (I_S). The level of motion is evaluated by processing the bi-dimensional convolution signal given by the following expression:

$$mot(x, y, t) = \sum_{i=1}^3 (\sum_{j=1}^3 H_{ij} C_{ij}) \quad (1)$$

where H_{ij} and C_{ij} are the elements of the following matrices:

$$C = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (2)$$

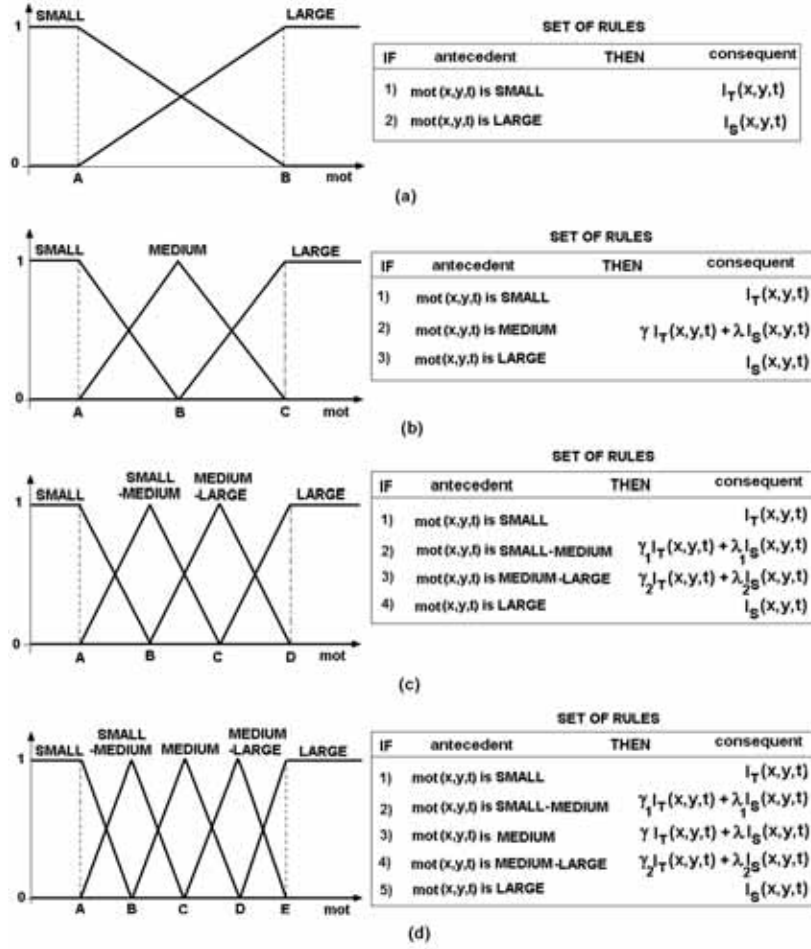


Fig. 1. Membership functions used by the different fuzzy inference systems

$$H = \begin{pmatrix} H(x-1, y-1, t-1) & H(x-1, y, t) & H(x-1, y+1, t-1) \\ H(x, y-1, t-1) & H(x, y, t) & H(x, y+1, t-1) \\ H(x+1, y-1, t-1) & H(x+1, y, t) & H(x+1, y+1, t-1) \end{pmatrix} \quad (3)$$

The notation (x,y,t) means that the pixel has the spatial coordinates (x,y) and corresponds to the instant (t) in the video sequence. Observing the size of the matrices H and C, a bi-dimensional convolution window of size 3x3 has been chosen. The idea of using bi-dimensional convolution techniques was introduced in [8]. Its main advantage is the inclusion of neighbors (with different weighting factors, as shown in expression (2)) to estimate the motion. In this case, the selected window allows to consider a spatio-temporal neighborhood. This could

reduce the eventual errors introduced by the presence of noise, edges or vertical details.

In the motion adaptive technique originally introduced in [5], the level of motion was evaluated by comparing the signal value corresponding to the luminance difference between two consecutive fields with a constant threshold value. The aim of the work presented in this paper is to improve the original motion adaptive technique by fuzzifying the levels of motion so as to perform a gradual instead of abrupt change between spatial and temporal interpolation. Therefore, in those areas where the level of motion is medium and, hence, the decision is not trivial, a non-linear interpolation between I_S and I_T is realized.

2.1 Fuzzy Inference System Description

The heuristic knowledge used by the motion adaptive techniques is modeled by employing a fuzzy inference system. Firstly, a system with two rules is used, where the concepts 'SMALL motion' and 'LARGE motion' are represented by the fuzzy sets of the Figure 1(a). Nevertheless, the interpolation capacity of fuzzy logic could be further exploited by considering the possibility of extending the number of fuzzy sets. In this sense, it is possible to define a new fuzzy set represented by the MEDIUM label shown in Figure 1(b). The set of rules is enlarged with a new rule that, when activated, performs a linear combination between the techniques I_S and I_T .

The level of motion in a field could not only be considered as SMALL, MEDIUM or LARGE, but more situations can be distinguished. For example four (SMALL, SMALL-MEDIUM, MEDIUM-LARGE, LARGE) or five labels (SMALL, SMALL-MEDIUM, MEDIUM, MEDIUM-LARGE, LARGE), represented in the Figure 1(c) and 1(d), could be employed. This translates into using four or five fuzzy 'if-then' rules, respectively. The problem when trying to implement these rules is that heuristic knowledge does not provide enough information to fix the constant values gamma and lambda of the rules' consequents neither to determine the values A, B, C, D, and E that describe the five possible linguistic labels. In order to fix these values, our approach has been to use supervised learning algorithms, as detailed in the following sections.

2.2 Supervised Learning Algorithm

The above described fuzzy systems have been implemented with the development environment Xfuzzy 3 [9]. This environment eases the design of fuzzy logic-based inference systems by including different CAD tools for the description, identification, simplification, verification, tuning and synthesis of the systems. In particular, the tool named *xfl* aids in the tuning stage, which is usually one of the most complex tasks in the design of fuzzy systems. This tool allows to apply different supervised learning algorithms where the desired behavior of the system is described by a set of training patterns. In our problem of video de-interlacing, the fuzzy systems have been tuned by using a set of progressive frames to generate

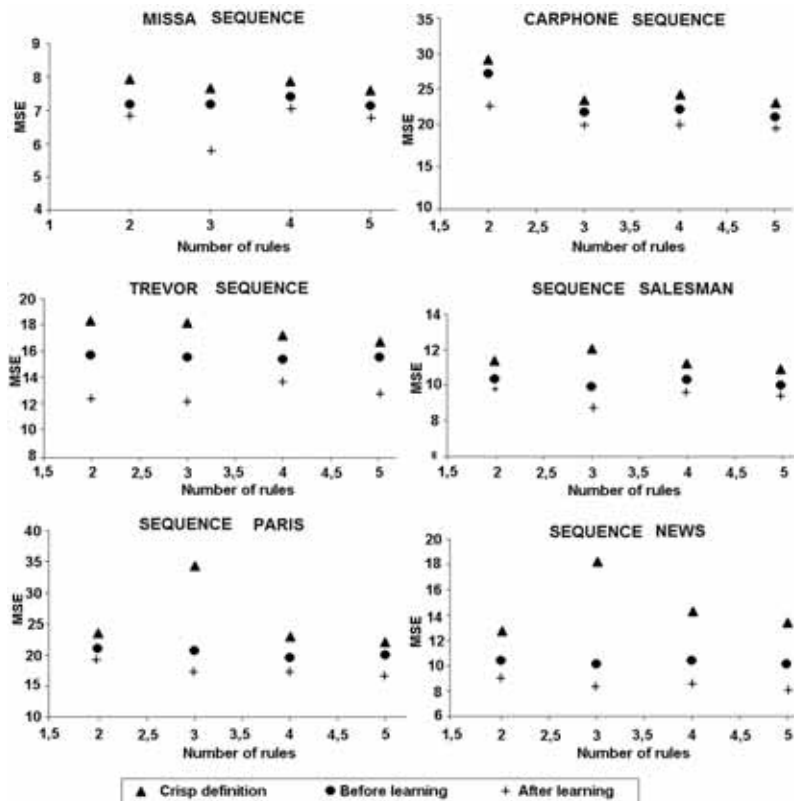


Fig. 2. MSE results obtained by the different fuzzy inference systems de-interlacing several video sequences

the training patterns. The selected supervised learning algorithm (Marquardt-Levenberg in our case) tries to minimize a function error which evaluates the difference between the current behavior and the desired one (determined by the input/output patterns). The tool *xfl* allows the user to select the parameters of the fuzzy system to be involved in the tuning process. The utility of this stage in the design process of fuzzy systems for de-interlacing video sequences is explained in detail in Section 3.

3 Simulation Results

In order to analyze the performance of the proposed fuzzy systems several benchmark video sequences have been considered. Since they are originally in a progressive format, a set of progressive images from all of these sequences have been selected to generate the training data for the supervised learning process. Afterwards, they have been de-interlaced artificially by eliminating lines from the frames.

Table 1. Average PSNR (values in dBs) when de-interlacing several video sequences

Sequence	Missa	Salesman	Carphone	Paris	Trevor	News
Format	CIF	CIF	QCIF	CIF	CIF	QCIF
LD	36.44	29.75	28.25	23.61	31.05	25.18
I_S	40.47	33.53	32.61	36.67	35.04	29.25
I_T	38.36	36.17	30.64	29.86	34.36	33.13
2fields-VT	40.25	36.54	34.08	30.73	36.61	35.46
3fields-VT	40.52	36.95	34.54	31.37	37.16	35.67
Technique in [7]	40.01	37.62	32.27	33.12	35.38	34.73
2-rule Proposal	40.18	38.29	34.78	35.28	36.69	37.51
3-rule Proposal	40.51	38.44	34.83	35.78	37.49	38.68
4-rule Proposal	39.65	38.23	34.94	35.5	36.77	38.65
5-rule Proposal	39.67	38.21	34.94	35.93	37.16	39.15

Figure 2 shows the mean squared error (MSE) value obtained when the artificially interlaced video sequences are de-interlaced. This value corresponds to the average MSE value of the de-interlaced fields (approximately fifty fields of each video sequence have been simulated). The graphics in Figure 2 illustrate the results obtained by an algorithm which uses a crisp definition of the concepts SMALL, SMALL-MEDIUM, MEDIUM, MEDIUM-LARGE and LARGE. They also show the results obtained when those concepts are defined as fuzzy sets and are processed by fuzzy systems with different number of rules (before and after learning). Comparing the three series of results, a first conclusion is that the use of fuzzy instead of crisp concepts provides lower errors. A second conclusion is that performance improves when the membership functions as well as the consequents are modified by the supervised learning algorithm. Finally, analyzing the number of rules and the MSE value, it can be observed that the system with three rules always obtains better results than the system with two ones. In the other side, the systems with four and five rules either do not provide significant improvement or even introduce a slightly higher number of errors.

The proposed fuzzy logic-based technique has been compared with: (a) basic linear techniques such as line doubling (LD), line average (I_S) as spatial technique and pixel insertion from the previous field (I_T) as temporal technique; (b) with linear spatio-temporal techniques [3], [4], which are currently used in commercial chips; and (c) with the fuzzy logic-based motion adaptive algorithm reported in [7]. Analyzing the results shown in Table 1, it can be seen how the highest results of PSNR, and hence the lowest errors, correspond to the proposed fuzzy systems (the results in Table 1 are obtained with the systems after learn-

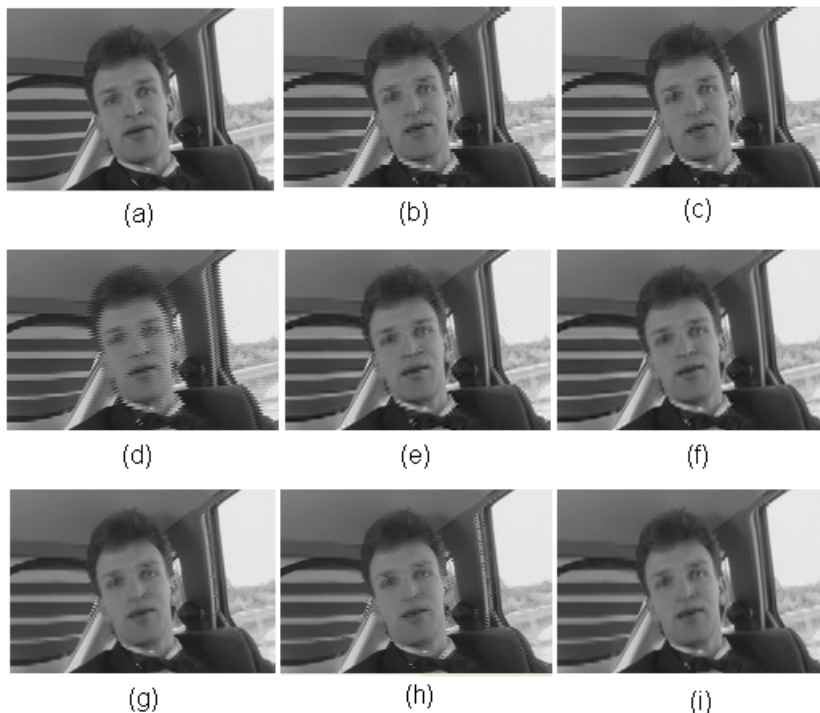


Fig. 3. (a) Progressive frame of 'Carphone' sequence. De-interlaced image applying: (b) line doubling, (c) line average, (d) field insertion, (e) 2-field VT filtering, (f) 3-field VT filtering, (g) fuzzy motion adaptive in [7], (h) proposal with 2 rules and (i) proposal with 3 rules

ing). The superior performance of our proposal can be also seen by analyzing the de-interlaced images in Figure 3.

Finally, an analysis of the computational cost involved in the implementation of each one of the proposed systems has been realized. All the algorithms have been executed on the same platform (a PC with a Pentium IV processor and the operating system MSWindow XP) so as to measure the time taken by each one in processing one sequence. The results are shown in Table 2. It can be seen how the linear techniques are the fastest ones but their results are widely improved by our proposal.

4 Conclusions

A fuzzy motion adaptive technique is presented in this paper. It performs a combination between two linear techniques depending on the level of motion. The proposal is inspired in the original motion adaptive idea, but it uses fuzzy definitions instead of crisp ones to describe the level of motion and employ a

Table 2. Time invested in de-interlacing fifty fields of a video sequence.

Algorithm	DL	I _S	I _T	VT 2fields	VT 3fields	Technique [7]	Proposal 2-3-4-5 rules
Time(s)	2.03	2.05	3.28	10.62	14.65	143.03	29.23-30.95-31.76-32.65

supervised learning technique to adjust the parameters of the fuzzy systems. After analyzing several systems with different complexity it can be concluded that a fuzzy inference system with three 'if-then' rules provides a good trade-off between performance and computational cost.

Acknowledgement. This work has been partially funded by the project TEC2005-04359/MIC from the Spanish Ministry of Education and Science, and TIC2006-635 from the Andalusian Regional Government. The first author is supported by the Spanish Ministry of Education under the program F.P.U for Phd. students.

References

1. G. De Haan. Video Processing. University Press, Eindhoven, 2004.
2. G. De Haan and E. B. Bellers. De-interlacing: An overview. *Proc. of the IEEE*, Vol. 86, Pág.1839-1857, 1998.
3. Genesis Microchip, Inc., Preliminary data sheet of Genesis gmVLD8, 8 bit digital video line doubler, version 1, 1996.
4. M. Weston. Interpolating lines of video signals. US-patent 4, Pág.789-893, 1998.
5. A. M. Bock. Motion adaptive standards conversion between formats of similar field rates. *Signal Processing: Image Communication*, Vol. 6, no. 3, Pág.275-280, 1994.
6. T. Doyle and M. Looymans. Progressive scan conversion using edge information. *Signal Processing of HDTV*. Ed. Elsevier Science Publishers, Vol. II, Pág.711-721, 1990.
7. D. Van de Ville, B. Rogge, W. Philips and I. Lemahieu. De-interlacing using fuzzy-based motion detection. *Proc. 3rd Int. Conf. on Knowledge-Based Intelligent Information Engineering Systems*, Pág.263-267, 1999.
8. J. Gutiérrez-Ríos, F. Fernández-Hernández, J. C. Crespo and G. Triviño. Motion adaptive fuzzy video de-interlacing method based on convolution techniques. *Proc. of Information Processing and Management of Uncertainty in Knowledge-Bsed Systems*, 2004.
9. F. J. Moreno-Velo, I. Baturone, S. Sánchez-Solano and A. Barriga. Rapid design of complex fuzzy systems with XFUZZY. *Proc. IEEE Int. Conf. on Fuzzy Systems*, Págs.342-347, 2003.
10. F. J. Moreno-Velo, I. Baturone, R. Senhadji and S. Sánchez-Solano. Tuning complex fuzzy systems by supervised learning algorithms. *Proc. IEEE Int. Conf. on Fuzzy Systems*, Págs. 226-231, 2003.